

Empirical Studies in Process Model Verification

Jan Mendling

Queensland University of Technology
126 Margaret Street, Brisbane QLD 4000, Australia
j.mendling@qut.edu.au

Abstract. Despite the large body of knowledge on formal analysis techniques for process models, in particular Petri nets, there has been a notable gap of empirical research into verification. In this paper we compare the few studies that report results from applying verification techniques to real-world process model collections. For this comparison we are particularly interested in the different approaches, their computational performance, and the number of errors found. Our comparison reveals that most of the samples have error rates of 10% to 20%. Some of the studies have established a connection between error probability and process model metrics, as well as between model understanding and both metrics and modeling competence of the model reader. Based on these results, we discuss implications and directions for future research.

1 Introduction

Even though workflow and process modeling have been used extensively over the past 30 years, we know surprisingly little about the factors that contribute to the quality of a process model and how respective quality assurance can be facilitated in real-world projects. This observation contrasts the large body of knowledge that is available on the formal analysis and verification of desirable properties, in particular for Petri nets. Furthermore, there is a notable disconnection between these mostly formal contributions and high-level conceptual work on guidelines and quality frameworks, e.g. [1–3]. Clearly, an empirical research agenda is needed to acquire new insights on quality and usage aspects of process modeling [4, 5] and its relationship to validation and verification (V&V).

It is a fundamental insight of software engineering that design errors should be detected as early as possible [6, 7, 4]. The later errors are detected, the more rework has to be done, and the more design effort has been at least partially useless. This also holds for the consecutive steps of analysis, design and implementation in the business process management life cycle [8, 9]. Yet, there are only a few papers that discuss validation and verification (V&V) as part of process design, e.g. [10], and the support for V&V activities is rather poor in current process modeling tools [11]. On the other hand, there are clearly quality issues with real-world process models. Recent studies report a significant rate of models with control flow errors in industry process model collections [11–15]. The mentioned lack of V&V features in tools as well as the lack of modeling competence in large-scale modeling projects [8] contribute to such high error rates.

These rates are particularly problematic when design models with undetected errors are forwarded to the implementation phase, to the frustration of system engineers. Therefore, the so-called gap between business process design and implementation phase, i.e. the limited reuse of conceptual process models in later design stages, might be partially caused by a lack of quality assurance in the early design phase.

Against this background, this paper aims to compare five recent studies on process model verification for real-world model collections, namely [11–15]. Such a comparison is beneficiary for the process verification community. It offers conclusions on the performance of verification approaches, though limited, and on the value of verification as a means of quality assurance. For instance, the decomposition approach used in [14] can be combined with reachability analysis in future research. Furthermore, the comparison is important for the process modeling community. The studies highlight issues with process modeling practice that have implications for the design activity and its tool support. Error patterns and their relative frequency allow us to derive guidelines that contribute to a less error-prone modeling style. The Seven Guidelines of Process Modeling (7PMG) [16] demonstrate that empirical insights can be directly fueled back into the design process.

The remainder of the paper is organized as follows. In Section 2 we introduce validation and verification as two complementary activities for quality assurance of process models. We discuss how both relate to each other and which techniques have been applied in verification studies. Section 3 presents findings from these verification studies and compares them in terms of error rates and performance. Building on these results, Section 4 discusses explanations of error rates and implications for process modeling. Finally, Section 5 concludes the paper.

2 The Scope of Verification

Validation and verification (V&V) are integral parts for establishing confidence in the quality of a process or a workflow model. Section 2.1 defines the scope of both validation and verification. Our focus will be the control flow of the process model. Section 2.2 introduces Event-driven Process Chains (EPCs), a modeling language that is considered in several verification studies, and define the notion of error. We use EPCs also because they include a superset of routing elements that are found in languages like BPMN, UML and YAWL. Furthermore, we illustrate typical verification issues by the help of an example. Finally, Section 2.3 presents verification techniques that have been used in recent empirical studies.

2.1 Validation versus Verification

The importance of V&V was recognized in software engineering from the start. As programming is in essence a problem-solving task, it implies that the validity of the solution must be established [17]. In this context, the IEEE Standard Glossary [18] defines Validation and Verification (V&V) as

“the process of determining whether the requirements of a system or component are complete and correct, the products of each development phase fulfill the requirement or conditions imposed by the previous phase, and the final system or component complies with specified requirements.”

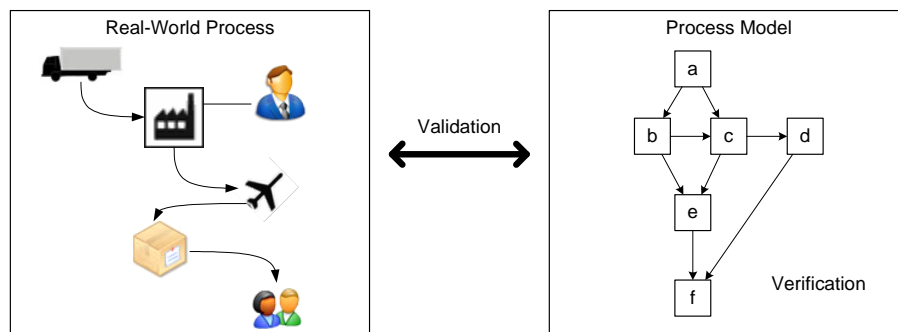


Fig. 1. Real-world process and process model

Different authors distinguish validation and verification as two complementary quality assurance steps [19–22].

Verification In essence, verification addresses both the general properties of a model and the satisfaction of a given formula by a model (see Figure 1). Related to the first aspect, formal correctness criteria play an important role in process modeling. Several criteria have been proposed including soundness for Workflow nets [23], relaxed soundness [24], or well-structuredness (see [25, 26] for comparisons). The second aspect is the subject of model checking and involves issues like separation of duty constraints, which can be verified, for example, by using linear temporal logic (LTL) [27]. The key characteristic of verification is that it relates to the internal correctness of a process model. Since it operates on the formal structure of the process model, it can be conducted without considering the real-world process.

Validation In contrast to that, validation addresses the consistency of the model with the universe of discourse, i.e. the real-world process (see Figure 1). As it is an external correctness criterion, it is more difficult and more ambiguous to decide. While verification typically relies on an algorithmic analysis of the process model, validation requires the consultation of the specification and discussion with process stakeholders. Although validation requires human judgement as a key characteristic, it should be noted that formal methods are useful to support it. For instance, simulation, animation or derivation of natural-language statements facilitate the validation of a process model by users.

The relationship between validation and verification in process modeling has been discussed in [10]. Most notably, the authors propose to perform first verification, then validation. In this way, validation is only conducted if verification succeeds. People involved with the real-world process have to be consulted if the verification problem cannot be resolved based on the existing documentation. If one of the steps points to problems, the model has to be modified appropriately followed by a new iteration of verification and validation. A consequence of this procedure is that validation will be conducted less often than verification. This speeds up the design time and reduces cost: since verification is performed automatically by the help of tools, it is cheaper and faster than validation.

2.2 Formal Errors of Process Models

Four of the recent verification studies [11–13, 15] use Event-driven Process Chain (EPC) business process models as input. Therefore, we briefly introduce EPCs and some potential correctness issues of their control flow. For formalization of EPC syntax and semantics refer to [28, 29].

The Event-driven Process Chain (EPC) is a business process modeling language for the representation of temporal and logical dependencies of activities in a business process [31]. EPCs offer *function type* elements to capture the activities of a process and *event type* elements describing pre- and post-conditions of functions. Furthermore, there are three kinds of *connector types* (i.e. AND, OR, and XOR) for the definition of complex routing rules. Connectors have either multiple incoming and one outgoing arc (join) or one incoming and multiple outgoing arcs (split). *Control flow arcs* are used to link elements.

The informal (or intended) semantics of an EPC can be described as follows. The AND-split activates all subsequent branches in a concurrent fashion. The XOR-split represents a choice between exclusive alternative branches. The OR-split triggers one, two or up to all of multiple branches based on conditions. In both cases of the XOR- and OR-split, the activation conditions are given in events subsequent to the connector. Accordingly, splits from events to functions are forbidden with XOR and OR since the activation conditions do not become clear in the model. The AND-join waits for all incoming branches to complete, then it propagates control to the subsequent EPC element. The XOR-join merges alternative branches. The OR-join synchronizes all active incoming branches, i.e., it needs to know whether the incoming branches may receive tokens in the future. This feature is called *non-locality* since the state of all (transitive) predecessor nodes has to be considered.

Figure 2 shows an EPC of the SAP Reference Model that was analyzed in a recent verification study [13]. This figure illustrates nine errors that were found. In essence, there are two types of errors that may occur in an incorrect process model: deadlocks and lack of synchronization. In the simplest case, a *deadlock* results from a combination of an XOR-split with an AND-join: while the split only activates one control path, the AND is waiting for both to be completed. A deadlock can be therefore described as a situation where synchronizing joins do not receive control from enough incoming branches to proceed. In the figure, the

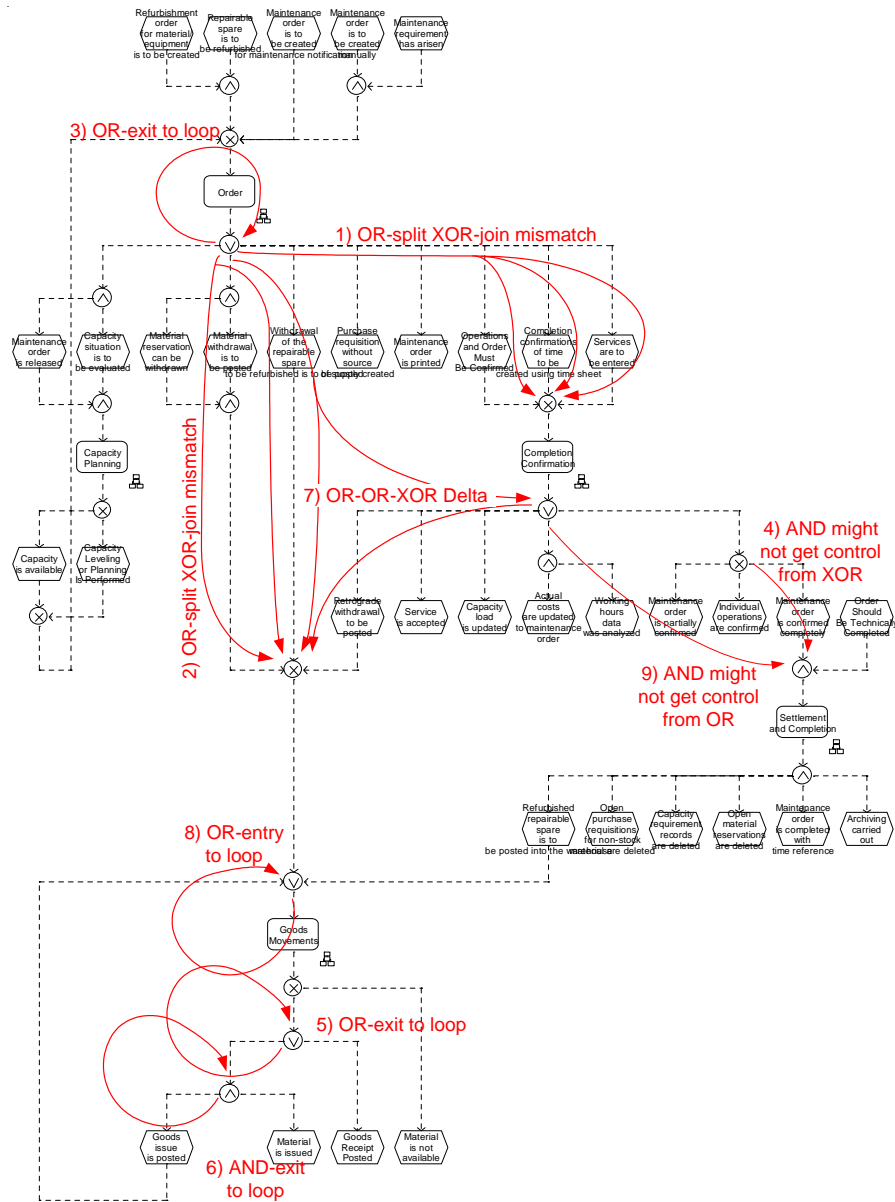


Fig. 2. The Refurbishment Processing in Plant Maintenance EPC from the SAP Reference Model [30, p.153] with errors being highlighted

errors 4 and 9 are potential deadlocks because the AND can receive control from one input branch but, in that case, it is not guaranteed that it also gets control from the other. Error 8 is a deadlock according to the semantics of [29] because the OR-join waits for itself via the branch from within the loop. A process model suffers from a *lack of synchronization* if multiple branches are activated, but not synchronized later at XOR-joins. Instances of this error type are errors 1, 2, 3, 5, 6, and 7 where multiple incoming branches can be active at the same time.

We define an *error* of a process model as a structural pattern that results in a deadlock or in a lack of synchronization. In order to identify errors, the formal behavior of the process has to be analyzed. This behavior is often defined based on *states and transitions*. Different *correctness criteria* can be verified using the graph of reachable states (*reachability graph*). They differ in computational complexity and in the precision of pinpointing errors. The following section presents the verification approaches of the five studies considered in this paper.

2.3 Verification Approaches

Verification is of particular importance to process modeling. Careless design can easily lead to process models with deadlocks or lack of synchronization which can have a serious business consequences if these models are used as workflows in operations. There are different aspects of a process model (e.g. control flow, data flow or resource allocation), and we focus on control flow here. The need for verification techniques stems from the fact that the complexity of the behavior specified in a process model is difficult to analyze for a human modeler [13]. This complexity is often referred to as the state explosion problem which is inherent to the specification of concurrency [19]. This problem imposes constraints on the applicability of state space analysis since the reachability graph can become exponentially large or even infinite. Therefore, reduction and decomposition techniques are used in order to improve the performance of verification. Below we discuss the correctness criteria that have been used in recent studies and the way they are implemented to verify industry-scale process models. First, we introduce soundness as the classical correctness criterion for process models as a reference. Then, relaxed soundness, an interactive verification approach, EPC soundness, single-entry-single-exit decomposition, and structuredness are discussed. These have been used in the five studies.

Soundness Soundness is an important and prominent correctness criterion for business process models and was first introduced in [23]. The original soundness property is defined for a Workflow net, a Petri net with one source and one sink, and requires that (i) for every state reachable from the source, there exists a firing sequence to the sink (option to complete); (ii) the state with a token in the sink is the only state reachable from the initial state with at least one token in it (proper completion); and (iii) there are no dead transitions [23]. It has been shown that soundness of a Workflow net is equivalent to liveness and boundedness of the corresponding short-circuited Petri net [23]. Therefore, several liveness and boundedness analysis techniques [32, 33]

are directly applicable to the verification of soundness. Tools like Woflan facilitate the verification of soundness in practice [34, 35]. *Soundness identifies all deadlocks and lack of synchronization for process models with one start and one end node.*

Relaxed Soundness The soundness property of workflow models has stimulated the specification of several soundness derivatives, mainly because some soundness aspects proved to be too restrictive in certain application domains. In [24] the authors argue that business processes are often conceptually modeled in such a way that only the desired behavior results in a proper completion. Since such conceptual models are not used for workflow execution, deadlocks are resolved by the people working in the process in a cooperative and ad-hoc fashion. The authors define a process to be relaxed sound if every transition in a Petri net representation of the process model is included in at least one proper execution sequence [24]. The relaxed soundness property is used in a study on the verification of the SAP Reference Model [36, 12]. For this study the authors automatically transform the 604 EPCs of the reference model to YAWL nets which are then analyzed using the WofYAWL tool. WofYAWL uses Petri nets analysis techniques including reduction rules and transition invariants to avoid calculating the whole state space [37]. *Relaxed soundness does not necessary identify all deadlocks and lack of synchronization.*

Interactive Verification In [38, 11] the authors introduce an interactive approach for the verification of EPCs. In a first step, reduction rules are automatically applied on the original EPC. Then, the user has to specify the meaningful combinations of start events that can initiate the process. After that, the EPC is translated to a Workflow net and the state space is generated. Then, the user decides for each final state whether it is intended or not. Given these pieces of information, the EPC is classified as correct if the Workflow net is sound, as maybe correct if the net is relaxed sound, and incorrect otherwise. The approach is applied to the procurement module of the SAP Reference Model. The verification is facilitated by an analysis plug-in for ProM [11]. *The interactive approach can identify all those deadlocks and lack of synchronization that occur for all start combinations.*

EPC Soundness Since EPCs may have multiple start and end events, the original soundness definition for workflow nets cannot be directly used for them. Therefore, the property of EPC soundness is proposed in [29]. It builds on the identification of a set of initial markings that covers all start events. The EPC soundness definition demands that (i) there exists such a non-empty set of initial markings, and (ii) that for each initial marking in this set proper completion is guaranteed. Furthermore, (iii) there must be a set of final markings reachable from some of these initial markings such that there exists at least one final marking in which a particular end arc holds a token. If that is fulfilled, proper completion is guaranteed for a set of initial markings that cover all start arcs. The EPC soundness property is stricter than the relaxed soundness criterion since it requires a guarantee of proper completion. Therefore, using it might reveal more errors. EPC soundness is used in

another verification study of the SAP Reference Model [13]. The authors use a two-step approach using first reduction rules and then, if necessary, state space calculation. *EPC soundness can identify all those deadlocks and lack of synchronization that occur for all start combinations in which a particular start event participates.*

Decomposition The approach reported in [14] builds on the decomposition of a workflow graph (essentially a free-choice Workflow net) into single-entry-single-exit (SESE) components. Such components can be calculated in linear time using depth-first search techniques [39]. The authors identify some heuristics for sound and for unsound components. In essence, these heuristics match EPC reduction rules as described in [30]. For their study, the authors use an implementation that reads process models in the format of the IBM WebSphere Business Modeler. The verification is conducted for a sample of 340 workflow graphs. *The decomposition approach identifies all those deadlocks and lack of synchronization that are defined in the heuristics.*

Reduction Another heuristic for checking the correctness of process models is used in [15]. The authors identify reduction rules for structured EPC models and heuristics to correct simple connector mismatch errors. The verification is performed for 285 EPCs from student projects, theses, text books, and scientific papers. *This approach identifies all those deadlocks and lack of synchronization that can be traced back to variants of unstructuredness.*

The different approaches have in common that they use reduction and decomposition techniques to address the state explosion problem. The performance of these approaches partially comes at the cost of precision, for instance, fast decomposition does not reveal all errors. While there is a complete set of reduction rules for free-choice Petri nets [40], none of the techniques used in the studies is complete. Therefore, reduction rules and state space analysis are often combined to balance performance and completeness [13, 11]. In practice, reduction is at least that powerful that the reduced process models can be analyzed with state space techniques [13].

3 Verification Results

In this section we discuss the results from the five verification studies. The samples used in these studies include:

1. *SAP Reference Model*: The development of the SAP reference model started in 1992 [41, p.VII] and continued until version 4.6 of SAP R/3, released in 2000 [42]. It includes 604 EPCs for documentation of the system.
2. *Service Model*: This collection of EPCs stems from a German process reengineering project in the service sector, conducted with academic supervision. The models that were defined in this project include 381 EPCs.
3. *Finance Model*: This collection contains the EPCs of a process documentation project in the Austrian financial industry. It includes 935 EPCs.

4. *Consulting Model*: This collection covers a total of 83 EPCs from three different consulting companies. The models are mainly used as reference models to support consulting activities of the companies.
5. *BPM Books*: This sample is built from four German textbooks on EPC process modeling [43–46] and includes 113 EPCs.
6. *Workflow 1 and Workflow 2*: These samples include IBM workflow models from industry projects [14]. The first includes 140 workflows, the second 200.
7. *Academia*: This collection includes documentation EPCs from student and industry projects from the University of Leipzig [15].

The different background of the samples highlights limitations of doing a comparison. Most notably, samples have been created for different purposes and using different languages. We would assume that workflow models were more rigorously defined and that restrictions in the IBM workflow language (one start, one end node) might provide less opportunities for errors. There are indeed low error rates with the workflow samples as Table 1 reveals. This table shows the number of models, the number of models that had errors, and the error rate for all samples. For some samples we have also indications of size (average number of arcs) and average processing time. We discuss the table below. In particular, we consider three aspects. Section 3.1 compares figures on the performance of the verification algorithm for the studies that report them. Section 3.2 analyzes the error rates of the different samples. Finally, Section 3.3 investigates explanations for the variation in error rates across the samples.

Table 1. Verification Studies

Sample	Ref.	Criterion	Models	Error Models	Error Rate	Average Arcs	Average Time (sec)
SAP RM	[12]	Relaxed Sound	604	34	5.6%	21	46.6
SAP Procurement	[11]	Interactive	40	4	10.0%		
SAP RM	[13]	EPC Sound	604	126	20.9%	21	1.8
Services	[13]	EPC Sound	381	37	9.7%		
Finance	[13]	EPC Sound	935	31	3.3%		
Consulting	[13]	EPC Sound	83	21	25.3%		
BPM Books	[13]	EPC Sound	113	25	21.4%		
Workflow 1	[14]	Decomposition	140	0	0.0%	67	0.06
Workflow 2	[14]	Decomposition	200	24	12.0%	126	0.08
Academia	[15]	Reduction	285	107	37.5%		

3.1 Verification Performance

Information on the performance of verification is only reported in a few studies. Table 1 summarizes the average number of arcs and the average processing time per model in seconds. The analyses reported in [12] and [13] operate on the same

sample, therefore they can be easily compared. The EPC soundness verification using reduction rules is about 20 times faster than the relaxed soundness analysis using WofYAWL. Still, it needs to be mentioned that the WofYAWL processing time includes the conversion of an EPC to a YAWL model by a command-line tool and the internal conversion of a YAWL net into a Petri net in WofYAWL. Without further information it is difficult to estimate which share the conversion contributes to overall processing. Still, it is reasonable that the simple element-wise conversion would not be too expensive. The slow processing might be explained by the inherent complexity of invariant calculation. WofYAWL uses invariant detection algorithms [47] that are exponential in space in the worst case. In contrast to that, the approach based on reduction rules benefits from reducing the models quite quickly. As reported in [30] that approach eliminates on average more than 91% of the model elements. Furthermore, it has a polynomial complexity since it uses reduction rules. In contrast to that, the approach of [14], that builds on calculating the program structure tree, is linear in complexity, see [39]. This fact is clearly visible in the relative performance: while the SAP reference model is analyzed by reduction in $1.8/21 = 0.09$ seconds per edge, the larger library 2 sample of [14] only requires $0.08/126 = 0.0006$ seconds per edge. This is about 150 times faster. Such performance figures clearly demonstrate the feasibility of improving performance of process model verification in practice.

3.2 Average Error Rates

The average error rates of the different samples in Table 1 give a good indication of how many errors can be expected in process model collections in practice. It must be mentioned that none of the verification approaches is complete. Accordingly, all figures have to be interpreted as lower bounds as non of the applied techniques guarantees all errors to be detected. Furthermore, the different approaches differ in precision: for example, the EPC soundness verification points to 126 errors in the SAP reference model while the relaxed soundness analysis only finds 34 problematic processes. Most of the approaches find a considerable amount of errors in the different samples: Library 1 has the lowest rate of 0%, followed by the Finance sample (3.3%) up to 37.5% in the Academia sample used in [15]. Altogether, from the 2741 distinct models of the studies 371 are not correct. This yields an error rate of 13.53%. The large variation in error rates raises the question why different samples are less error-prone. In the next section we discuss some of the factors.

3.3 Variation in Error Rates

Based on cognitive considerations like bounded rationality [48] and limited information processing capabilities of humans [49], it has been hypothesized that the understanding of a process model has a significant impact on error probability [30]. In the process of constructing a model, understanding relates to two factors: the structure of the model and the modeling capabilities of the modeler. These

two factors have been investigated in prior research, namely [12, 13, 50–52]. In the following we summarize the findings.

Different sets of structural metrics of process models have been analyzed for their potential to explain and predict error probability [12, 13, 52]. The authors use logistic regression as a statistical tool. This way, it can be estimated how the value of a particular metric influences error probability. The relaxed soundness study [12] uses a set of simple count metrics (number of functions, events, and connectors of different type) and the SAP reference model as a sample for the estimation. It is shown that a substantial part of the variation can be explained as measured by the Nagelkerke R^2 value of about 30% as a coefficient of determination. In [13] the authors extend this work by using a larger sample of about 2000 models (SAP reference model plus the Service, Finance, and Consulting sample) and a set of more sophisticated ratio metrics including structuredness, connector heterogeneity, and depth. The resulting model explains more than 90% of the variation as measured by the Nagelkerke R^2 value. Accordingly, the fact whether a process model contains errors seems to be largely influenced by its size, its degree of structuredness and similar metrics.

The experiments reported in [50] and [51] investigate both model structure and competence of the model reader as determinants for model understanding. The questionnaires of these experiments use different process models and ask the participants questions that reveal whether the content of the model is interpreted correctly. The first experiment is conducted on paper with a sample of 73 graduate students from Eindhoven University of Technology, the University of Madeira, and the Vienna University of Economics and Business Administration [50]. It shows that the average degree of connectors is negatively correlated to understanding and that those students being trained in Petri nets concepts perform better. The second experiment was conducted as an online questionnaire. It was filled out by 46 respondents [51]. The result show a strong correlation between theoretical process modeling knowledge and understanding of the different models. From the structural metrics the degree of separability,¹ i.e. how easy can the model be separated in two disconnected parts, had a strong correlation with understanding. Apparently, the simpler models were understood better. It is interesting to note that in both experiments participants tended to overestimate their ability to correctly read the models. This confirms earlier observations [49].

These findings suggest that errors do not occur arbitrarily, but in specific constellations. This has several implications as the following section discusses.

4 Implications

In this section we discuss implications of the verification studies. In particular, Section 4.1 identifies implications for the way business process modeling is conducted in practice. Then, Section 4.2 highlights some facts that tool vendors should consider. Finally, Section 4.3 points to some directions of future research.

¹ This is the ratio of cut vertices to all nodes of the process model

4.1 Implications for Process Modeling

The results of the different studies confirm that process models in practice suffer from quality problems. Furthermore, the studies suggest that industry process model collections are likely to have error rates of 10% to 20%. Clearly, there are differences in error rates, and there are different structural metrics that are closely connected with error probability. Based on these connections and on work on activity labeling [53], Mendling, Reijers, and Van der Aalst propose a set of seven process modeling guidelines (7PMG) that are supposed to direct the modeler to creating understandable models that are less prone to errors [16].

Table 2. Seven Process Modeling Guidelines [16]

G1	Use as few elements in the model as possible
G2	Minimize the routing paths per element
G3	Use one start and one end event
G4	Model as structured as possible
G5	Avoid OR routing elements
G6	Use verb-object activity labels
G7	Decompose a model with more than 50 elements

Table 2 summarizes the 7PMG guidelines. Each of them is supported by empirical insight into the connection of structural metrics and errors or understanding. The size of the model has undesirable effects on understandability and likelihood of errors [50, 12, 13]. Therefore, **G1** recommends to use as few elements as possible. **G2** suggests to minimize the routing paths per element. The higher the degree of elements in the process model the harder it becomes to understand the model [50, 13]. **G3** demands to use one start and one end event, since the number of start and end events is positively connected with an increase in error probability [13]. Following **G4**, models should be structured as much as possible. Unstructured models tend to have more errors and are understood less well [13, 15, 52, 50]. **G5** suggests to avoid OR routing elements, since models that have only AND and XOR connectors are less error-prone [13]. **G6** recommends using the verb-object labeling style because it is less ambiguous compared to other styles [53] Finally, according to **G7** models should be decomposed if they have more than 50 elements.

These guidelines have to be considered as directions for achieving a good verification result. They are founded on the insight that there are alternative ways of expressing the same behavior in a process model. Clearly, a rework of a process model should not affect the validation goal. Creating small models might help to achieve several guidelines, for instance **G1**, but it reduces the precision of the model. Therefore, the guidelines should be followed if the validity is not reduced.

4.2 Implications for Tool Vendors

The amount of errors in the model samples from practice emphasizes the importance of verification. Most of the samples have error rates between 10% and 20%. While verification has been discussed for some time, the studies demonstrate that different approaches can handle large sets of several hundred business process models on a common desktop computer. In particular, the performance of the decomposition approach is that good (linear time) that a continuous verification would be possible every time the designer modifies parts of a model.

While the verification techniques are apparently mature enough to deal with large models from practice, there seems to be too little attention paid to verification issues by tool vendors. Indeed, tool vendors should be interested in these techniques since the lack of respective features has a negative impact on the productivity of the business process modeling exercise: models cannot be reused for system development, business users cannot interpret the models properly, and conclusions can hardly be drawn from the models regarding process performance. Building on validation and verification features, the tool vendors can easily provide a greater benefit to their customers and help to improve the process of designing business process models.

4.3 Implications for Future Research

The results of the five studies have several implications of future research. In this section we focus on some of them, in particular, error explanation and auto-repair, derivation of behavior-equivalent models, novel verification techniques, and concepts for process model quality assurance.

Explanation The Petri net analyzer Woflan already emphasizes the need for re-design assistance in case the verification of a model fails [35]. The approaches of the studies we considered address this issue as well. The reduction rule approach for EPC soundness generates SVG graphics of the process model with errors being highlighted (cf. Figure 2). The work on structuredness proposes an auto-repair for certain error patterns like connector mismatch in structured blocks [15]. The problem with such an approach is that there are usually at least two options to repair a model: changing the split behavior or changing the join behavior. Which of them is applicable must be decided by the domain expert who knows the real-world process. In a more sophisticated subgraph structure errors can be caused by the combination of multiple splits and multiple joins. Recommendations for repairing such structures would be a valuable contribution of future research.

Behavior-Equivalent Models Some work has been done on the creation of behavior-equivalent process models. Most notable is the work on Petri net synthesis from a labeled transition system [54, 55]. This work has been used to eliminate OR-joins from process models [56]. Further research discusses the untangling of unstructured loops and the derivation of structured BPEL from arbitrary Petri nets [57, 58]. A desirable contribution would be to

change a process model in such a way that the behavior is kept, but the structure optimized, e.g. along the lines of the seven process modeling guidelines. Designers would highly appreciate such a feature that would contribute to more understandable models.

Verification Techniques In the comparison we have seen the power of the program structure tree decomposition. To our best knowledge this technique has not yet been applied to the analysis of Petri nets. The EPC soundness study highlights the benefit of combining reduction and reachability graph analysis. A similar approach can be taken to increase the precision of the program structure tree verification.

Quality Assurance Validation and verification techniques are discussed in different neighboring disciplines of process modeling, namely requirement engineering [59], knowledge-based systems [60], simulation [61], or conceptual modeling [62]. Some validation and verification techniques for process models are discussed in [10, 9]. Yet, an overarching framework and a systematic analysis of how suitable of such techniques from other areas are for process modeling is missing.

Despite these technical challenges, the analysis of process models is a field where commercial tool support is lagging behind academic concepts. There is a need for a closer collaboration between academia and industry in order to make verification techniques available in a broader range of tools.

5 Conclusions

In this paper we discussed verification of process models and findings from empirical studies on it. Most of the approaches use some kind of decomposition technique, or a combination of reduction and reachability analysis. The best performance shows the linear time decomposition approach based on the program structure tree. The precision of the different approaches is difficult to compare since different model collections are used and a benchmark verification sample is not available. The comparison of the studies reveals that most of the samples have error rates of 10% to 20%. Some of the studies have established a connection between error probability and process model metrics, as well as between model understanding and both metrics and modeling competence of the model reader. These findings have implications for the way process models are constructed. In this context, we discussed the seven process modeling guidelines. The high error rates are a clear signal to tool vendors to add verification features to their tools. Finally, there are still several challenges for future work. In particular, we identified a need for further research into error repair, derivation of behavior-equivalent models, combined verification techniques, and quality assurance for process models.

References

1. Lindland, O., Sindre, G., Sølvyberg, A.: Understanding quality in conceptual modeling. *IEEE Software* **11** (1994) 42–49

2. Becker, J., Rosemann, M., Uthmann, C.: Guidelines of Business Process Modeling. In van der Aalst, W., Desel, J., Oberweis, A., eds.: *Business Process Management. Models, Techniques, and Empirical Studies*. Springer, Berlin et al. (2000) 30–49
3. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action: a revised quality framework. *Europ. J. of Inf. Systems* **15** (2006) 91–102
4. Moody, D.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowl. Eng.* **55** (2005) 243–276
5. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? *Data & Knowl. Eng.* **58** (2006) 358–380
6. Boehm, B.: *Software Engineering Economics*. Prentice-Hall (1981)
7. Wand, Y., Weber, R.: Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research* **13** (2002) 363–376
8. Rosemann, M.: Potential pitfalls of process modeling: part a. *Business Process Management Journal* **12** (2006) 249–254
9. Philippi, S., Hill, H.: Communication support for systems engineering - process modelling and animation with april. *Journal of Sys. & Softw.* **80** (2007) 1305–1316
10. van Hee, K., Sidorova, N., Somers, L., Voorhoeve, M.: Consistency in model integration. *Data & Knowledge Engineering* **56** (2006)
11. van Dongen, B., Vullers-Jansen, M., Verbeek, H., van der Aalst, W.: Verification of the sap reference models using epc reduction, state-space analysis, and invariants. *Computers in Industry* **58** (2007) 578–601
12. Mendling, J., Verbeek, H., van Dongen, B., van der Aalst, W., Neumann, G.: Detection and Prediction of Errors in EPCs of the SAP Reference Model. *Data & Knowl. Eng.* **64** (2008) 312–329
13. Mendling, J., Neumann, G., van der Aalst, W.: Understanding the occurrence of errors in process models based on metrics. In Meersman, R., Tari, Z., eds.: *OTM Conference 2007, Proceedings, Part I*. LNCS 4803 (2007) 113–130
14. Vanhatalo, J., Völzer, H., Leymann, F.: Faster and more focused control-flow analysis for business process models through sese decomposition. In Krämer, B., Lin, K.J., Narasimhan, P., eds.: *Service-Oriented Computing - ICSOC 2007, Proceedings*. LNCS 4749 (2007) 43–55
15. Gruhn, V., Laue, R.: What business process modelers can learn from programmers. *Science of Computer Programming* **65** (2007) 4–13
16. Mendling, J., Reijers, H., van der Aalst, W.: Seven Process Modeling Guidelines (7PMG). *Qut eprint, Queensland University of Technology* (2008)
17. Adrion, W., Branstad, M., Cherniavsky, J.: Validation, verification, and testing of computer software. *ACM Computing Surveys* **14** (1982) 159–192
18. IEEE: IEEE Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology . IEEE Computer Society Press (1990)
19. Valmari, A.: The state explosion problem. In Reisig, W., Rozenberg, G., eds.: *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets*. LNCS 1491 (1998) 429–528
20. Hoppenbrouwers, S., Proper, H., van der Weide, T.: A fundamental view on the process of conceptual modeling. In Delcambre, L., Kop, C., Mayr, H., Mylopoulos, J., Pastor, O., eds.: *Conceptual Modeling - ER 2005, Proceedings*. LNCS 3716 (2005) 128–143
21. Boehm, B.W.: *Software engineering; R & D trends and defense needs*. In: *Research Directions in Software Technology*. MIT Press (1979)
22. Sommerville, I.: *Software Engineering*. 6th edn. Addison-Wesley (2001)
23. van der Aalst, W.: Verification of Workflow Nets. In Azéma, P., Balbo, G., eds.: *Application and Theory of Petri Nets*. LNCS 1248 (1997) 407–426

24. Dehnert, J., Rittgen, P.: Relaxed Soundness of Business Processes. In Dittrick, K., Geppert, A., Norrie, M., eds.: Proceedings of the 13th International Conference on Advanced Information Systems Engineering. LNCS 2068 (2001) 151–170
25. Dehnert, J., Zimmermann, A.: On the suitability of correctness criteria for business process models. In van der Aalst, W., Benatallah, B., Casati, F., Curbera, F., eds.: Business Process Management, Proceedings. LNCS 3649 (2005) 386–391
26. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag (2007)
27. Pnueli, A.: The Temporal Logic of Programs. In: Proceedings of the 18th IEEE Annual Symposium on the Foundations of Computer Science (1977) 46–57
28. Kindler, E.: On the semantics of EPCs: Resolving the vicious circle. *Data & Knowledge Engineering* **56** (2006) 23–40
29. Mendling, J., van der Aalst, W.: Formalization and Verification of EPCs with OR-Joins Based on State and Context. In Krogstie, J., Opdahl, A., Sindre, G., eds.: Proceedings of CAiSE 2007. LNCS 4495 (2007) 439–453
30. Mendling, J.: Detection and Prediction of Errors in EPC Business Process Models. PhD thesis, Vienna University of Economics and Business Administration (2007)
31. Keller, G., Nüttgens, M., Scheer, A.W.: Semantische Prozessmodellierung auf der Grundlage “Ereignisgesteuerter Prozessketten (EPK)”. Heft 89, Institut für Wirtschaftsinformatik, Saarbrücken, Germany (1992)
32. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE **77** (1989) 541–580
33. Desel, J., Esparza, J.: Free Choice Petri Nets. Volume 40 of Cambridge Tracts in Theoretical Computer Science. Cambridge Univ. Press, Cambridge, UK (1995)
34. Verbeek, H., van der Aalst, W.: Woflan 2.0: A Petri-net-based Workflow Diagnosis Tool. In Nielsen, M., Simpson, D., eds.: Application and Theory of Petri Nets 2000. LNCS 1825 (2000) 475–484
35. Verbeek, H., Basten, T., van der Aalst, W.: Diagnosing Workflow Processes using Woflan. *The Computer Journal* **44** (2001) 246–279
36. Mendling, J., Moser, M., Neumann, G., Verbeek, H., van Dongen, B., van der Aalst, W.: Faulty EPCs in the SAP Reference Model. In S. Dustdar, J.F., Sheth, A., eds.: Proceedings of BPM 2006. LNCS 4102 (2006) 451–457
37. Verbeek, H., van der Aalst, W., ter Hofstede, A.: Verifying workflows with cancellation regions and or-joins: An approach based on relaxed soundness and invariants. *The Computer Journal* **50** (2007) 294–314
38. van Dongen, B., Jansen-Vullers, M.H.: Verification of SAP reference models. In van der Aalst, W., Benatallah, B., Casati, F., Curbera, F., eds.: Business Process Management, Proceedings. LNCS 3649 (2005) 464–469
39. Johnson, R., Pearson, D., Pingali, K.: The program structure tree: Computing control regions in linear time. In: Proceedings of the ACM SIGPLAN’94 Conference on Programming Language Design and Implementation (PLDI). SIGPLAN Notices **29**(6). (1994) 171–185
40. Esparza, J.: Reduction and synthesis of live and bounded free choice petri nets. *Information and Computation* **114** (1994) 50–87
41. Keller, G., Teufel, T.: SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping. Addison-Wesley (1998)
42. Mendling, J., van der Aalst, W., van Dongen, B., Verbeek, H.: Referenzmodell: Sand im Getriebe - Webfehler. *iX - Magazin für Professionelle Informationstechnik.* (in German) (2006) 131–133
43. Becker, J., Schütte, R.: Handelsinformationssysteme. 2nd edn. Moderne Industrie, Landsberg/Lech (2004)

44. Scheer, A.W.: Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. 2nd edition edn. Springer-Verlag (1998)
45. Seidlmeier, H.: Prozessmodellierung mit ARIS. Vieweg Verlag (2002)
46. Staud, J.: Geschäftsprozessanalyse: Ereignisgesteuerte Prozessketten und Objektorientierte Geschäftsprozessmodellierung für Betriebswirtschaftliche Standardsoftware. 3rd edn. Springer-Verlag (2006)
47. Colom, J., Silva, M.: Convex geometry and semiflows in P/T nets, A comparative study of algorithms for computation of minimal P-semiflows. In Rozenberg, G., ed.: Advances in Petri Nets 1990. LNCS 483 (1990) 79–112
48. Simon, H.: Sciences of the Artificial. 3rd edn. The MIT Press (1996)
49. Burton-Jones, A., Meso, P.: How Good are these UML Diagrams? An Empirical Test of the Wand and Weber Good Decomposition Model. In Applegate, L., Galliers, R., DeGross, J., eds.: Proceedings of ICIS. (2002) 101–114
50. Mendling, J., Reijers, H., Cardoso, J.: What makes process models understandable? In Alonso, G., Dadam, P., Rosemann, M., eds.: Business Process Management, Proceedings. LNCS 4714 (2007) 48–63
51. Mendling, J., Strembeck, M.: Influence factors of understanding business process models. In Abramowicz, W., Fensel, D., eds.: Proc. of the 11th International Conference on Business Information Systems (BIS 2008). LNBIP 7 (2008) 142–153
52. Laue, R., Mendling, J.: The impact of structuredness on error probability of process models. In Kaschek, R., Kop, C., Steinberger, C., Fliedl, G., eds.: Information Systems and e-Business Technologies: 2nd International United Information Systems Conference, UNISCON 2008, Proceedings. LNBIP 5 (2008)
53. Mendling, J., Reijers, H.: How to define activity labels for business process models? In Oberweis, A., Hesse, W., eds.: Proc. of the Third AIS SIGSAND Europe 2008. LNI (2008)
54. Ehrenfeucht, A., Rozenberg, G.: Partial (Set) 2-Structures - Part 1 and Part 2. Acta Informatica **27** (1989) 315–368
55. Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A.: Deriving petri nets from finite transition systems. IEEE Transactions on Computers **47** (1998) 859–882
56. Mendling, J., van Dongen, B., van der Aalst, W.: Getting Rid of the OR-Join in Business Process Models. In: Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC'07). (2007) 3–14
57. Zhao, W., Hauser, R., Bhattacharya, K., Bryant, B., Cao, F.: Compiling business processes: untangling unstructured loops in irreducible flow graphs. Int. Journal of Web and Grid Services **2** (2006) 68–91
58. van der Aalst, W., Lassen, K.: Translating unstructured workflow processes to readable BPEL: Theory and implementation. Inf.& Softw. T. **50** (2008) 131–159
59. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. (2000) 35–46
60. Tsai, W., Vishnuvajjala, R.: Verification and Validation of Knowledge-Based Systems. IEEE Transactions on Knowledge and Data Engineering **11** (1999) 202–212
61. Sargent, R.: Verification and validation of simulation models. (2005) 130–143
62. Frederiks, P., van der Weide, T.: Information modeling: The process and the required competencies of its participants. Data & Knowl. Eng. **58** (2006) 4–20